UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/723,391	11/25/2003	Marcus Felipe Fontoura	ARC920030080US1	8873

7590 09/05/2008

Frederick W. Gibb, III
McGinn & Gibb, PLLC
Suite 304
2568-A Riva Road
Annapolis, MD 21401

EXAMINER	
WONG, JOSEPH D	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
09/05/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/723,391
Filing Date: November 25, 2003
Appellant(s): FONTOURA ET AL.

Peter A. Balnave, Ph. D.
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 17 June 2008 appealing from the Office action mailed 25 January 2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is generally correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

The following is a listing of the evidence (e.g., patents, publications, Official Notice, and admitted prior art) relied upon in the rejection of claims under appeal.

(A) US Pre-Grant Publication 2003/0159112 A1, Pub. Date 21 August 2003, hereinafter Fry.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1, 13, 25, 37, 38-53 are rejected under 35 U.S.C. 102(e) as being anticipated by Fry, (US 2003/0159112), hereinafter Fry.

Regarding claim 1, Fry teaches a method for (interpreted to be an intended use) query processing by using a streaming application programming interface (API) for a mark-up language data stream of a textual document (see Abstract, cover Fig., wherein the textual limitation is met text shown in Fig. 3), said method comprising: producing, by said streaming API for a mark-up language data stream (Abstract, wherein a mark-up language data stream is interpreted to include an "XML stream"), an ordered index of all textual elements corresponding to their order in said mark-up language data stream (Fig. 3, wherein the elements are shown in an order), said ordered index comprising tag identifiers and end positions corresponding to each of said all textual elements ([30, 35], wherein navigation to the end of a document is discussed, also see claim 1); scanning by a processor (Fig. 1, item 106 meets the limitation), all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers ([31], wherein the ordered index is stepped until the element to be extracted is processed by the base parser); parsing a matched textual element ([31], supra), if (interpreted to be conditional) a tag identifier ([31], wherein the tag identifier is interpreted to be an "element

tag"), corresponding to said matched textual element, matches said query ([31], wherein iterative stepping conditioned on a match meets the limitation, [11, 18, 21, 28-29, 32, 35]); and skipping an unmatched textual element for parsing ([28-30], wherein the skip function allows the programmer to "skip ahead to specific sections...or get subsections'"), if (interpreted to be conditional) a tag identifier, corresponding to said unmatched textual element, does not match said query. ([28-30], wherein "Element type two ends when another end tag is reached in the document", also see Fig. 3)

Regarding claim 13, Fry teaches a system for query processing by using a streaming application programming interface (API) for a mark-up language data stream of a textual document (see Abstract, cover Fig., wherein the textual limitation is met text shown in Fig. 3), said system comprising: an ordered index of all textual elements corresponding to their order in said mark-up language data stream (Abstract, wherein a mark-up language data stream is interpreted to include an "XML stream"), said ordered index comprising tag identifiers and end positions corresponding to each of said all textual elements and being produced by said streaming API for a mark-up language data stream (Fig. 3, wherein the elements are shown in an order); a processor (Fig. 1, item 106 meets the limitation) adapted to scan all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers ([30, 35], wherein movement to the end of a document is discussed, also see claim 1); and a parser adapted to parse a matched textual element ([31], supra), if a tag identifier corresponding to said matched textual element ([31], wherein iterative stepping conditioned on a match meets the limitation, [11, 18, 21, 28-29, 32, 35]), matches said query, and to skip an

unmatched textual element for parsing ([28-30], wherein the skip function allows the programmer to "skip ahead to specific sections...or get subsections"), if (interpreted to be conditional) a tag identifier, corresponding to said unmatched textual element, does not match said query. ([28-30], wherein "Element type two ends when another end tag is reached in the document", also see Fig. 3)

Regarding claim 25, Fry teaches a program storage device readable by computer comprising a program of instructions executable by said computer to perform a method for query processing by using a streaming application programming interface (API) for a mark-up language data stream of a textual document (see Abstract, cover Fig., wherein the textual limitation is met text shown in Fig. 3), said method comprising: producing, in said streaming API for a mark-up language data stream (Abstract, wherein a mark-up language data stream is interpreted to include an "XML stream"), an ordered index of all textual elements corresponding to their order in said mark-up language data stream (Fig. 3, wherein the elements are shown in an order), said ordered index comprising tag identifiers and end positions corresponding to each of said all textual elements (Fig. 3, wherein the elements are shown in an order); scanning by a processor (Fig. 1, item 106 meets the limitation), all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers; parsing a matched textual element, if a tag identifier (interpreted to be conditional), corresponding to said matched textual element ([31], wherein iterative stepping conditioned on a match meets the limitation, [11, 18, 21, 28-29, 32, 35]), matches said query; and skipping an unmatched textual element for parsing ([28-30], wherein the skip function allows the programmer to "skip ahead to

specific sections...or get subsections'"), if a tag identifier (interpreted to be conditional), corresponding to said unmatched textual element, does not match said query. ([28-30], wherein "Element type two ends when another end tag is reached in the document", also see Fig. 3)

Regarding claim 37, Fry teaches a system for a query processing by using a streaming application programming interface (API) for a mark-up language data stream of a textual document (see Abstract, cover Fig., wherein the textual limitation is met text shown in Fig. 3), said system comprising: an ordered index of all textual elements corresponding to their order in said mark-up language data stream (Abstract, wherein a mark-up language data stream is interpreted to include an "XML stream"; Fig. 3, wherein the elements are shown in an order), said ordered index comprising tag identifiers and end positions corresponding to each of said all textual elements and being produced by said streaming API for a mark-up language data stream (see Abstract, Fig. 3); a processor (Fig. 1, item 106 meets the limitation) adapted to scan all tag-identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers ([31], wherein iterative stepping conditioned on a match meets the limitation, [11, 18, 21, 28-29, 32, 35]); and a parser adapted to skip an unmatched textual element, if (interpreted to be conditional) a tag identifier , corresponding to said unmatched textual element [31], supra), does not match said query. ([28-30], wherein "Element type two ends when another end tag is reached in the document", also see Fig. 3)

Regarding claim 38, Fry teaches the method, all the limitations of which are incorporated herein by reference, wherein each of said textual elements comprises textual sub-

elements, and each of said textual sub-elements comprised a tag identifier and an end position. ([28-31], Figs. 2-3)

Regarding claim 39, Fry teaches the method, all the limitations of which are incorporated herein by reference, further comprising storing a parsed matched textual element in a buffer. ([28-31], Figs. 2-3)

Regarding claim 40, Fry teaches the method, all the limitations of which are incorporated herein by reference, wherein upon said skipping of said unmatched textual element for parsing, said method further comprises offsetting said parser based upon an end position stored in said ordered index and corresponding to said unmatched textual element. ([28-31], Figs. 1-3, limitation is interpreted to be a negative)

Regarding claim 41, Fry teaches the method, all the limitations of which are incorporated herein by reference, wherein said mark-up language data stream comprises HyperText Markup Language (HTML) (interpreted to be an optional alternative) or Extensible Markup Language (XML). (Abstract)

Regarding claim 42, Fry teaches the system, all the limitations of which are incorporated herein by reference, wherein each of said textual elements comprises textual sub-elements, and each of said textual sub-elements comprising a tag identifier and an end position. ([28-31], Figs. 2-3)

Regarding claim 43, Fry teaches the system, all the limitations of which are incorporated herein by reference, further comprising a buffer, operatively connected to said parser, in which a parsed matched textual element is stored. ([28-31], Figs. 1-3)

Regarding claim 44, Fry teaches the system, all the limitations of which are incorporated herein by reference, wherein said parser is offset based upon an end position stored in said ordered index that corresponds to said unmatched textual element. ([28-31], Figs. 2-3)

Regarding claim 45, Fry teaches the system, all the limitations of which are incorporated herein by reference, wherein said mark-up language data stream comprises Hyper Text Markup Language (HTML) (interpreted to be an optional alternative) or Extensible Markup Language (XML). (Abstract)

Regarding claim 46, Fry teaches the program storage device, all the limitations of which are incorporated herein by reference, wherein each of said textual elements comprises textual subelements, and each of said textual sub-elements comprises a tag identifier and an end position. ([28-31], Figs. 1-3)

Regarding claim 47, Fry teaches the program storage device, all the limitations of which are incorporated herein by reference, further comprising storing a parsed matched textual element in a buffer. ([28-31], Figs. 1-3)

Regarding claim 48, Fry teaches the program storage device, all the limitations of which are incorporated herein by reference, wherein upon said skipping if said unmatched textual element for parsing, said method further comprises offsetting said parser based upon an end position stored in said ordered index and corresponding to said unmatched textual element. ([28-31], Figs. 1-3, limitation is interpreted to be a negative)

Regarding claim 49, Fry teaches the program storage device, all the limitations of which are incorporated herein by reference, wherein said mark-up language data stream comprises Hyper Text Markup Language (HTML) (interpreted to be an optional alternative) or Extensible Markup Language (XML). (Abstract)

Regarding claim 50, Fry teaches the system, all the limitations of which are incorporated herein by reference, wherein each of said textual elements comprises textual sub-elements, and each of said textual sub-elements comprising a tag identifier and an end position. ([28-31], Figs. 1-3)

Regarding claim 51, Fry teaches the system, all the limitations of which are incorporated herein by reference, further comprising a buffer, operatively connected to said parser, in which a parsed matched textual element is stored. ([28-31], Figs. 1-3)

Regarding claim 52, Fry teaches the system, all the limitations of which are incorporated herein by reference, wherein said parser is offset based upon an end position stored in said ordered index that corresponds to said unmatched textual element. ([28-31], Figs. 1-3, limitation is interpreted to be a negative)

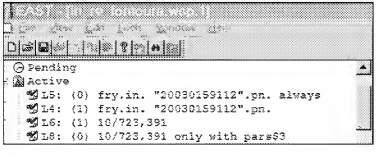
Regarding claim 53, Fry teaches the system, all the limitations of which are incorporated herein by reference, wherein said mark-up language data stream comprises Hyper Text Markup Language (HTML) (interpreted to be an optional alternative) or Extensible Markup Language (XML). (Abstract)

(10) Response to Argument

Appellant argues that Fry contrasts with the present invention. However, the Examiner will show that these contrasts are merely Appellant's characterizations that are shown to be less than fully consistent with the prior art of Fry, the instant claims, instant specification and rule. Furthermore, the Examiner will show that Appellant's selectively applied reasoning would be more fair and equitable if reapplied among the prior art of Fry, the instant specification and instant claims. Finally, the Examiner will request the Board to sustain the rejection of instantly pending claims..

On page 12, paragraph 3, Appellant argues "Fry **always** parses the extracted, located, and read element type, but he may not read the sub-elements". In response to Appellant's argument that the reference shows a certain feature in contrast with Appellant's invention, it is noted that the feature upon which Appellant relies (i.e., "**always**") is not expressly recited in the prior art of

Fry. Furthermore, Fry has another means of locating other than parsing such as using "pointing to different positions in a document" in Fry, paragraph [32] which enables skipping an intervening portion between the positions. Attempts at arguing "always parses" appear subtly misleading with respect to the "**skip**ping" over portions of Fry that are unfavorable to Appellant's instant claim 1 such as Fry's Abstract, last sentence, also paragraphs 11, 18, 21, 28. Therefore the prior art rejection of Fry should be sustained.



> Fry paragraph [0031] recites: In a method for utilizing such an event stream, the name of an element to be extracted from the XML document is passed to an iterative method built on top of a base parser, such as a SAX API or DOM parser. An element tag of the XML document is located and the element type read by the base parser, without necessarily reading the sub-elements or text of that element. The elements of the XML document are then stepped through by the base parser in combination with the iterative method until the element to be extracted is located, read, and processed by the base parser. An event is generated, that is related to the element, and placed on an event stream for use by an application such as a Java application

On page 11, paragraph 5, Appellant describes the present invention as "tag identifiers are scanned to determine whether a match exists between any of the tag identifiers and the query, parsing a matched textual element **only if** the tag identifier matches the query, and skipping parsing of the unmatched textual element, when the tag identifier does not match the query". In response to Appellant's argument that the references fail to show certain features of applicant's

invention, it is noted that the features upon which applicant relies (i.e., "**only if**") are not expressly recited in instantly rejected claims 1, 13, 25, and 37.

Furthermore, Fry teaches the argued portion of claim 1 parsing a matched textual element (paragraph [31], wherein iterative stepping conditioned on a match meets the limitation), if (interpreted to be a conditional skip in paragraphs [11, 18, 21, 28-29, 32, 35]) a tag identifier ([31], wherein the tag identifier is interpreted to be an "element tag"), corresponding to said matched textual element, matches said query ([31], wherein iterative stepping conditioned on a match meets the limitation, wherein the directly abutting portion is skipped in paragraphs [11, 18, 21, 28-29, 32, 35]); and skipping an unmatched textual element for parsing ([28-30], wherein the skip function allows the programmer to "skip ahead to specific sections...or get subsections'"), if (interpreted to be conditional) a tag identifier, corresponding to said unmatched textual element, does not match said query. ([28-30], wherein "Element type two ends when another end tag is reached in the document", also see Fig. 3)

On page 11, paragraph 5, Appellant's arguments state that parsing only happens if the tag matches the query. However this seems less than consistent with instant figure 1, item 130, "selectively skipping portions of the document *based on instructions from the index*" which states that the skipped portions are based on instructions from the index. Appellant does not discuss how to produce an index from a document without parsing at least the indexed portion of the document before or during or after the index is produced. This argument that the index step somehow skips parsing could distract the reader because the neither claim 1 nor the instant figures (especially figure 1) is silent on the question as if or how the index of the document is produced without parsing.

Furthermore, Appellant's description of the present invention is "parsing a matched textual element **only if** the tag identifier matches the query" is not supported within the instant specification. However, Appellant's description contrasts in that he word "parsing" appears added to the characterization of instant specification paragraph [11], sentence 5 and paragraph [40], sentence 5 refer to "indexing only" and instant claims 1, 13, 25, and 37 refer to "parsing" with the "**only if**". Consequently, Appellant's characterization of the instant claims and instant specification and any dependent arguments are shown to be moot because Appellant has argued qualifications not shown to be in the instant claims and instant specification. Therefore the prior art rejection of Fry should be sustained.

On page 12, paragraph 1, Appellant argues that Fry does not teach "matching a query to a tag identifier of a mark-up language data stream". However, Fry teaches in paragraph [31], that "the elements of the XML document are then stepped through by the base parser in combination with the iterative method until the element to be extracted is located". Locating a specific element is a determination achieves the correspondence effect of a matching step as also seen in paragraph [28], such that the streaming parser also allows the programmer to stop processing the document, *skip* ahead to specific sections of the document, and/or get subsections of the document as mini DOM trees. Again, Appellant's argument contradicts the Fry publication's explicit correspondence of "doc" to "<doc>" tag in paragraph [29], 'event "doc" which corresponds to the...<doc> tag in the XML document'. Therefore the rejections of claim 1, 13, 25 and 37 using Fry should be sustained.

On page 12, paragraph 2, Appellant alleges a contrast in skipping the parsing step by of an unmatched textual element, if a tag identifier, correspond to the unmatched textual element,

does not match the query. However, the Fry reference teaches skipping in many paragraphs. Examples include the last sentence of the abstract, "skipping unwanted XML from the document", paragraph [11], "the streaming API can also allow the programmer to stop processing the document, skip ahead to sections of the document, and get subsections of the document as mini-DOM trees. Paragraph [18] teaches "XML parser can provide a variety of types of access to the application...does not require the entire document to be read into memory, including providing an XML data stream, pulling XML information, and skipping unwanted XML from the document". Further skipping is seen in paragraphs [28-29], "allows the programmer to stop processing the document, skip ahead to specific sections of the document, and/or get subsections of the document as mini DOM trees. startElement ( ) is called and passed with the event "doc" which corresponds to the "<doc>" tag. Paragraph [32] recites "the stream can be thought of as pointing to different positions" a document". Again, Appellant's argument is contrary to the "skipping" capabilities explicitly shown in the Fry publication. Therefore the rejections of claim 1, 13, 25 and 37 using Fry should be sustained.

On page 12, paragraph 3, this is discussed initially.

On page 13, paragraph 4 through page 14, paragraph 1, Appellant alleges that Fry does not teach the limitations of claims 1, 13, 25 and 37 as a whole. Appellant's arguments are simply inconsistent with 37 CFR 1.111(c) because they do not clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. Further, they do not show how the amendments avoid such references or objections. Therefore the prior art rejections of claims 1, 13, 25 and 37 using Fry should be sustained.

Appellant reasons that Fry **"always parses"** without addressing the question of inherency and providing sufficient evidence of inherency as being limited to Fry while ignoring an "skip" capabilities in Fry. Fry paragraph [39] states that there are different plug in parser components of DOM, SAX, and more parsers with other behaviors that can be used so it seems premature of Appellant to dictate how Fry must parse. A full and fair argument from Appellant would apply Appellant's "always parses" test to an unclaimed feature of instant claim 1 in instant figure 1, item 100 , "producing at least one index of a document..." produces this index without at least parsing the indexed portion of the XML document during this step.. The reason why the indexed portion must be parsed is that there is reference and referent portion in an index and delineating the location of these necessarily involves parsing to build an index of a document. Therefore the Board can equitably reapply Appellant's "always parses" argument toward instant figure 1, item 100 because Appellant's means for skipping as specified is using an index from Fig. 1.

Appellant's reasons when arguing claim 1 above that the instant claim 1 "**parses only if**" without expressly reciting this phrase in instant claim 1. A full and fair argument from Appellant would explain how "skipping unwanted XML" capabilities or pointing to another location in the document skips parsing too. Therefore the Board can equitably reapply Appellant's "only parses" argument toward instant figure 1, item 100 (when producing an index) and Fry's Abstract, last sentence, or Fry's paragraphs 11, 18, 21, 28, 31-32.

The Examiner countered that Appellant's arguments are shown to have inconsistencies with the evidence of Fry as prior art, the instant claims, instant specification and rule. For at least the above reasons, it is believed that the rejections on all instantly appealed claims should be sustained.

(11) Related Proceedings Appendix

None.

Respectfully submitted,

/JOSEPH D WONG/ JDW

Asst. Examiner, Art Unit 2166

Conferees:

/Hosain T Alam/

Supervisory Patent Examiner, Art Unit 2166

/Vincent F. Boccio/ VFB
Primary Examiner, Art Unit 2169
Appeal Specialist TC2100